CLAIMS

1. A method of recovering from loading invalid data into a register within a pipelined processor, the method comprising the steps of:

(A) setting a register status for said register to an invalid state in response to loading invalid data into said register; and

(B) stalling said processor in response to an instruction requiring data buffered by said register and said register status being in said invalid state.

2. The method of claim 1, further comprising the step of:

(C) setting said register status for said register to said invalid state in response to a conditional store for said register.

3. The method of claim 1, further comprising the step of:

(D) setting said register status for said register to said invalid state in response to receiving data from another register having said invalid state.

4. The method of claim 1, further comprising the steps of:

(E) obtaining valid data for said register from a memory in response to loading invalid data into said register;

(F) stalling said processor in response to obtaining valid data for said register; and

(G) loading said register with valid data in response to stalling said processor in step (F).

5. The method of claim 4, further comprising the steps of:

(H) stalling said processor prior to writing new data to said register having said invalid state while obtaining said valid data for said register; and

(I) writing new data to said register in response to loading said register with valid data.

6. The method of claim 1, further comprising the step of:

(J) buffering said register status as a plurality of bits to provide for multiple conditions that would set said register status to said invalid state.

7. The method of claim 1, further comprising the steps of:

(C) setting said register status for said register to said invalid state in response to a conditional store for said register;

(D) setting said register status for said register to said invalid state in response to receiving data from another register having said invalid state;

(E) obtaining valid data for said register from a memory in response to loading invalid data into said register;

(F) stalling said processor in response to obtaining valid data for said register; and

(G) loading said register with valid data in response to stalling said processor in step (F).

8. A pipelined processor comprising:

a register configured to buffer (i) data and (ii) a register status; and

logic configured to (i) set said register status to an invalid state in response to loading invalid data into said register, and (ii) stall said processor in response to an instruction requiring data buffered by said register and said register status being in said invalid state.

9. The pipelined processor of claim 8 further comprising:

said logic being further configured to (iii) set said register status to said invalid state in response to a conditional store for said register.

10. The pipelined processor of claim 8, further comprising:

another register; and

said logic being further configured to (iv) set said register status to said invalid state in response to receiving data from said another register having said invalid state.

11. The pipelined processor of claim 8, further comprising:

a bus interface unit configured to obtain valid data for said register from a memory in response to loading invalid data into said register; and

said logic being further configured to (v) stall said processor in response to obtaining valid data for said register, and (vi) load said register with valid data in response to stalling said processor.

12. The pipelined processor of claim 11, further comprising:

said logic being further configured to (vii) stall said processor prior to writing new data to said register while said bus interface unit is obtaining valid data for said register, and (viii) write new data to said register in response to loading said register with valid data obtained from said memory.

13. The pipelined processor of claim 8, further comprising:

said register status being buffered as a plurality of bits to provide for multiple conditions that would set said register status to said invalid state.

14. The pipelined processor of claim 8, further comprising:

another register;

a bus interface unit configured to obtain valid data for said register from a memory in response to loading invalid data into said register; and

said logic being further configured to (iii) set said register status to said invalid state in response to a conditional store for said register, (iv) set said register status to said invalid state in response to receiving data from said another register having said invalid state, (v) stall said processor in response to obtaining valid data for said register, (vi) load said register with valid data in response to stalling said processor,

and (vii) set said register status to said invalid state in response to a cache load-miss for said register.

15. A pipelined processor comprising:

means for buffering data;

means for buffering a register status;

means for setting said register status to an invalid state in response to loading invalid data into said means for buffering data; and

means for stalling said processor in response to an instruction requiring data buffered by said means for buffering data and said register status being in said invalid state.